

（深圳）

实验报告

开课学期： 2025春季

课程名称：计算机组成原理（实验）

实验名称： 高速缓存器设计

实验性质： 设计型

实验学时： 4 地点：

学生班级： 20233119

学生学号： 2023313409

学生姓名： 房效民

作业成绩：

实验与创新实践教育中心制

2025年4月

|  |
| --- |
| 1、Cache详细设计 |
| 要求：绘制ICache的状态转换图，并结合关键代码，**详细描述**状态转移关系、转移条件、各状态的输入输出信号以及需要完成的操作。*\*若完成了附加题，则分别绘制DCache的读、写状态转换图，并配以文字详细描述相应的内容。* |
| IDLE：初始状态，当接受到inst\_valid信号后变成TAG\_CHK,检查标志位。状态  TAG\_CHK:该阶段维护hit信号，若检查到cache\_line\_r的tag部分和tag\_from\_cpu相同，说明命中，hit置1，根据offset返回计算返回偏移的指令。  若未命中，hit为0，则准备向主存发送读请求。进入REFILL状态  首先要检测dev\_rrdy是否为1，即是否可以向主存发送读信号。  若dev\_rrdy为1，cpu\_addr发送数据，cpu\_ren使能。信号持续一个周期，期间一直为REFILL状态。  当总线返回数据，dev\_valid置1同时dev\_rrdy置1，检测到dev\_valid后，状态从REFILL变成TGA\_CHK，在REFILL状态变成TAG\_CHK状态后的那个时钟周期，无时钟延迟写入cache块，则此时又变成了命中时的处理逻辑      有关cache块的写入与读出的逻辑如上：  读出：检测valid位和tag位，一致则读出数据，不一致则不做处理  写入：当cache\_we信号有效时写入，其中ran\_num通过随机数生成决定替换块。 |
|  |

|  |
| --- |
| 2、调试报告 |
| 要求：结合仿真波形截图对ICache作详细的时序分析，要求包含读命中、读缺失2种情形，每种情形列举1个测试用例**详细分析**。分析过程参考实验原理中的时序解读，但需**把模块内部关键信号添加到波形，并结合信号的实际取值**进行分析。*\*若完成了附加题，则需额外给出DCache的仿真波形截图及其详细文字分析，要求包含写命中、写缺失和Uncached访问3种情形，每种情况列举1个测试用例详细分析。* |
| 此处是访问命中的情形分析：  信号解析：  Inst\_rreq:来自cpu，访存使能  Inst\_addr:来自cpu访存地址  Inst\_valid:cache输出，信号是否有效  Inst\_out:cache输出：取到的指令  Dev\_rrdy：来自总线，是否可以进行取指令  Cpu\_ren:cache输出，是否开始发送读请求  Cpu\_raddr：读请求地址  Dev\_rvalid:来自总线，是否取到数据并且返回数据信号  Dev\_rdata:来自总线：总线返回的数据块  State:状态机，0为IDLE,1为TAG\_CHK,2为REFILL状态  Hit：命中信号  Offset:偏移量  当inst\_rreq有效，这时在cache中找到数据块，直接返回  Inst\_rreq持续一个周期，下一个时钟周期变成TAG\_CHK状态，hit有效，返回数据  下一个时钟周期，复位成IDLE状态    此处是访问不命中的情形分析：  信号解析：  Inst\_rreq:来自cpu，访存使能  Inst\_addr:来自cpu访存地址  Inst\_valid:cache输出，信号是否有效  Inst\_out:cache输出：取到的指令  Dev\_rrdy：来自总线，是否可以进行取指令  Cpu\_ren:cache输出，是否开始发送读请求  Cpu\_raddr：读请求地址  Dev\_rvalid:来自总线，是否取到数据并且返回数据信号  Dev\_rdata:来自总线：总线返回的数据块  State:状态机，0为IDLE,1为TAG\_CHK,2为REFILL状态  Hit：命中信号  Offset:偏移量  当inst\_rreq有效，这时在cache中没有找到数据块，当在TAG\_CHK阶段时，hit为0，因此在下一个时钟周期进入REFILL阶段，这时dev\_rrdy为高，发送总线读指令，cpu\_ren为高，cpu\_raddr为需要读取的总线地址**（由于取值是取一个数据段，所以offset的部分要全部变成0），这里比较坑。折磨笔者较长时间**。  接下来是漫长的等待总线阶段。。。。。。。    Dev\_valid标志读总线的结束，同时，dev\_rdata是总线返回的数据。这是时钟周期的下一个周期，状态变为tag\_chk,在dev\_valid为高的那一个周期，cache块完成了读与替换操作（这里具体可以看我代码）。因此在TAG\_CHK阶段，就如同之前的正常读命中一样处理，至此，一个完整的cache命中缺失读数据完成。 |

|  |
| --- |
| 3、思考与讨论 |
| 分别给出无ICache时和有ICache时，SoC运行测试程序的总时间的截图，并谈谈你对该测试结果的理解。 |
| 可见，使用icache可以极大地缩短取指令的时间。  其实观察测试数据，可以发现很多指令都是+4的顺序取指，中间穿插一些小的跳转，因此cache可以利用局部性原理降低访存的时间 |